SlideSpace

Automatic slide generation with Evolutionary Layout Optimization

Supervised by : DR. Amr S. Ghoneim

June 2025

Motivation

Professors and instructors often spend hours ***manually*** converting course materials or research papers into slide decks for lectures.

Employees and professionals must ***frequently*** turn technical documents or reports into clear, engaging presentations for stakeholders.

Manually transforming documents into clear, well-designed slides is a time-consuming process that demands both **deep understanding** and **visual design skills**

About the problem

*The technical problem is automating the **extraction**, **summarization**, and **visual structuring** of **multimodal content** from **unstructured documents** into slide presentations.*

01

Document parsing

Extract content (text, images, etc.) from various document formats

02

Summarization

Summarize dense text into clear, slide-ready points while preserving context

03

Multimodality

Handle multimodality by aligning visual elements with their related text

04

Layout Optimization

Optimize slide layout for clarity, balance, and visual appeal automatically

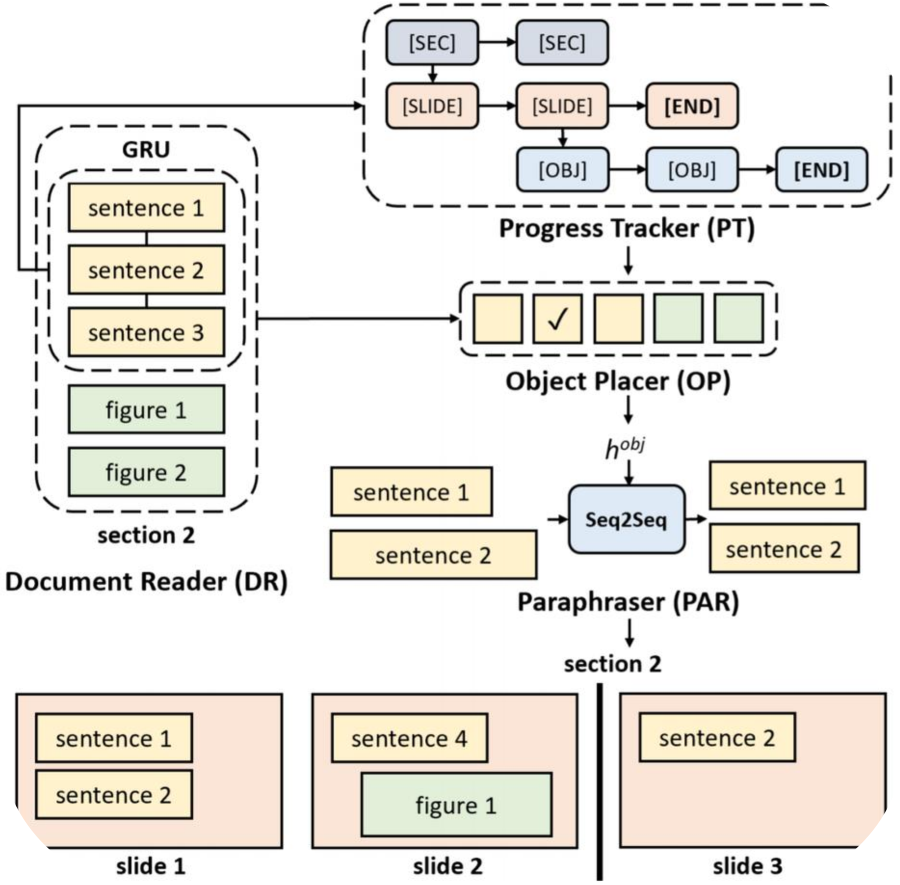
DOC2PPT approach & drawbacks

DOC2PPT (supervised) is an end-to-end model that generates slides from multimodal documents by hierarchically selecting and placing content in slides.

The dataset's focus on paper-slide pairs limits the model's generalization to non-academic documents.

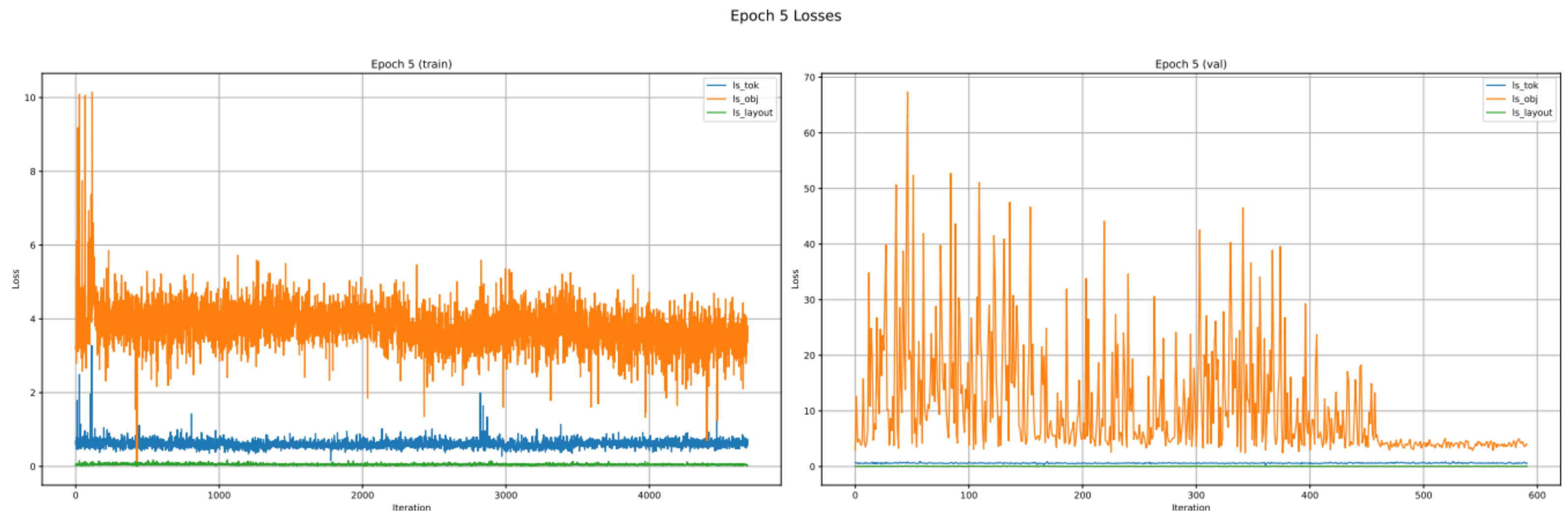
The system is not truly modular tightly coupled components with shared states hinder isolation, replacement, and independent tuning.

Their extraction tools rely on deprecated dependencies with discontinued support, leading to compatibility issues and reduced long-term maintainability

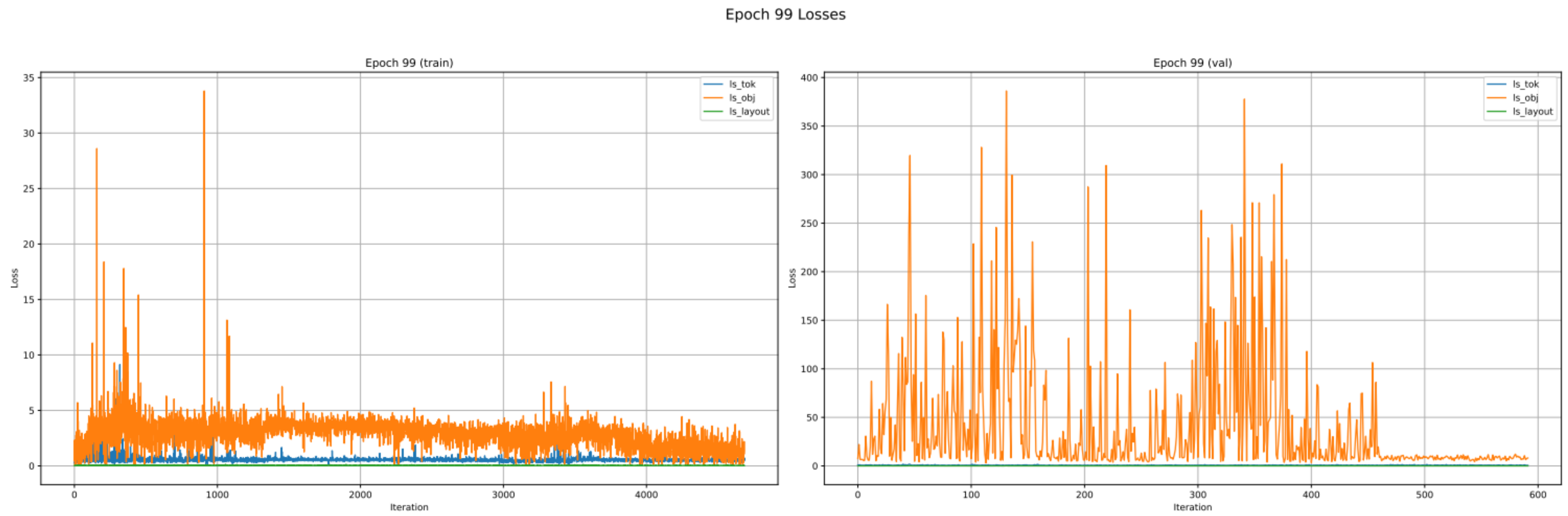


DOC2PPT approach & drawbacks

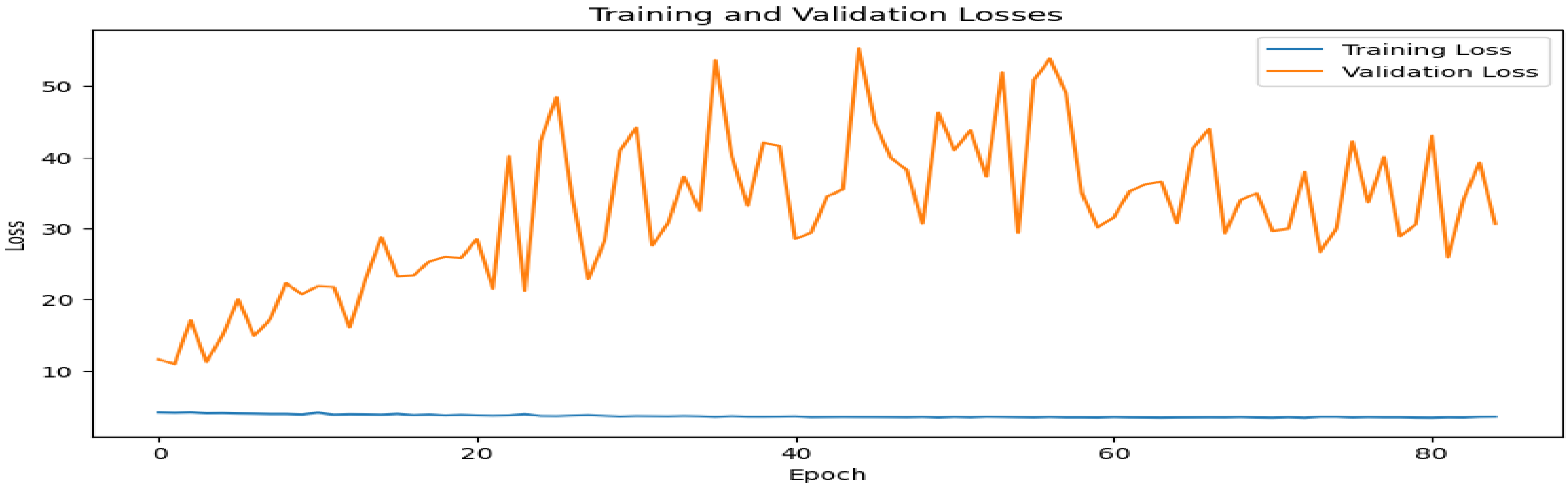
Epoch 5



Epoch 99



DOC2PPT approach & drawbacks



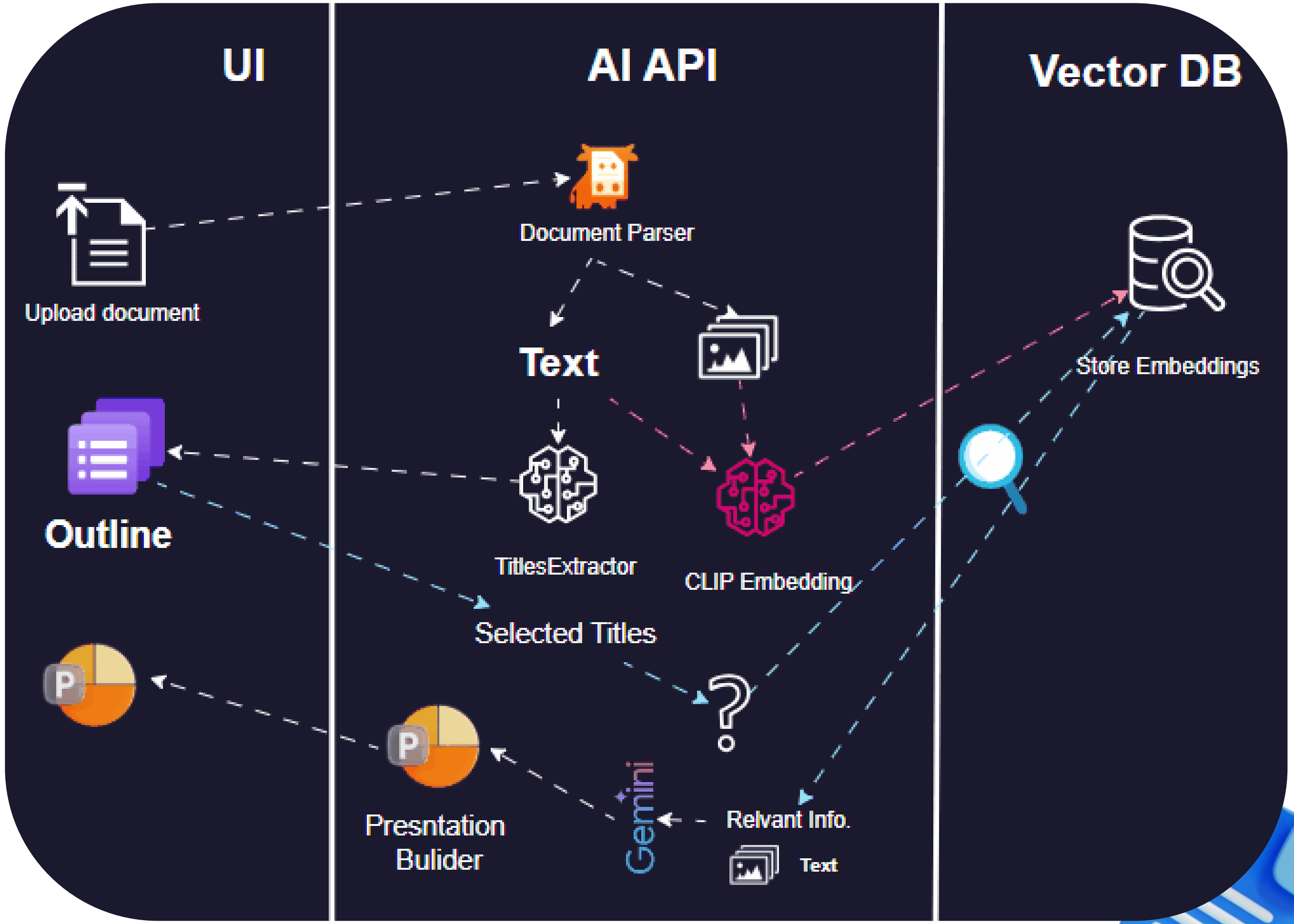
The training loss remained nearly flat, resembling a constant function, while the validation loss continuously increased suggesting the model failed to learn meaningful representations and is overfitting

Architecture Overview

UI – Web interface for uploading documents, select titles, and exporting presentations

AI API – Summarizes and structures document content using RAG-based NLP

ELO – Optimizes slide layout using evolutionary algorithms for clarity and design



Document Parsing

Documents are parsed based on their format (PDF or DOCX), extracting text and images using specialized tools like PyMuPDF, pdfplumber, and python-docx

The chunking uses recursive splitting with hierarchical separators (\n\n, ., etc.,) maintaining semantic boundaries while respecting a target chunk size

Chunks include overlap to preserve context across boundaries, enabling better coherence during retrieval.

Each chunk is labeled with metadata such as page/paragraph number and position, ensuring traceability and alignment with the original document.

Efficient storage & retrieval

Aspect	Traditional DB (SQL / NoSQL)	Vector DB (ChromaDB, FAISS, Qdrant)
Data Type	**Structured** (tables, JSON, key-value, etc.)	**Unstructured** High-dimensional vectors (embeddings)
Query Type	Exact match, range, join, filter	Similarity search (ANN: cosine, dot product, L2)
Indexing	B-tree, hash index, inverted index	IVF, HNSW, PQ, disk-based ANN indexes
Search Goal	Retrieve records that match exact conditions	Retrieve records most similar to a given query vector
Use Case	Traditional CRUD, business data, analytics	Semantic search, recommendation, NLP, image/audio retrieval
Scalability	Scales with rows and columns	Scales with dimensions and number of vectors
Multimodal Support	No (text/image must be preprocessed externally)	Yes stores and searches image, audio, and text embeddings

Efficient storage & retrieval

Aspect	Flat Embedding Search	Indexed Vector Search (e.g. IVF in ChromaDB)
Search Type	Exact linear scan (brute-force)	Approximate Nearest Neighbor (ANN)
Speed	Slower as dataset grows (O(n))	Much faster using partitioned/graph-based structures
Accuracy	High (exact distances)	Slightly lower (approximate), tunable trade-off
Indexing	No index — full scan each time	Uses structures like IVF, HNSW, PQ
Memory Usage	Simple, but scales poorly with large data	More efficient for large datasets
Use Case	Small datasets or high-precision search	Large-scale, real-time search across many vectors

Titles as Queries !

We extract the titles from the uploaded document by one of three strategies

Font-based

Detect titles by identifying text with the largest font sizes in the document

Semantic

Uses text embeddings to extract diverse and informative titles from the full document

Candidate

Filters structured lines and extracts ranked keyphrases as concise, relevant titles

Aspect	Font-Based	Semantic	Candidate
Titles	Medium	High	Low–Medium
Speed	Fast	Slow	Moderate
Precision	High (if styled)	Medium	High
Errors	Fonts inconsistent	Semantic drift	Structure-sensitive
Best for	Styled PDFs	Raw text / OCR	Semi-structured PDFs

Titles as Queries !

We transform short section titles into detailed, natural language questions using a language model (e.g., Flan-T5) trained for instruction following (optional).

This reformulation adds semantic depth to vague or generic titles like “Results” or “Training,” producing context-rich queries like *“How was the model trained?”* or *“What were the key findings?”*

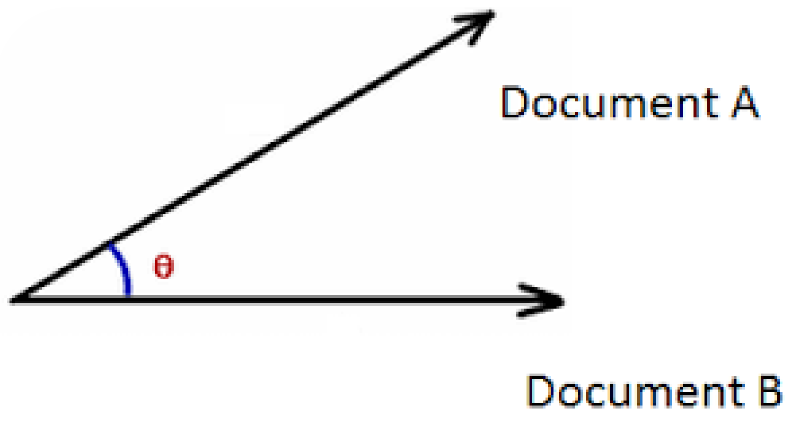
Using QA-style queries improves retrieval accuracy, as vector databases respond better to complete, meaningful sentences than isolated keywords.

It reduces semantic drift by making the intended focus of the query explicit, avoiding matches to unrelated text.

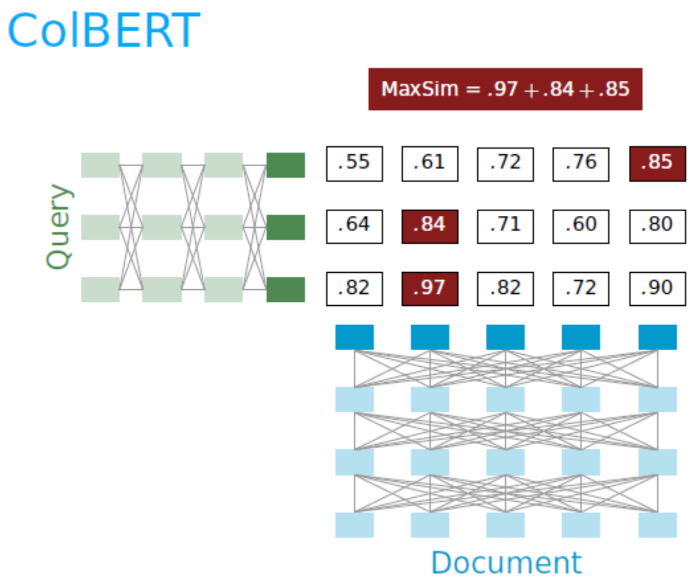
They also align better with answer-like document chunks, enhancing semantic similarity and content relevance.

Retrieving with reranker

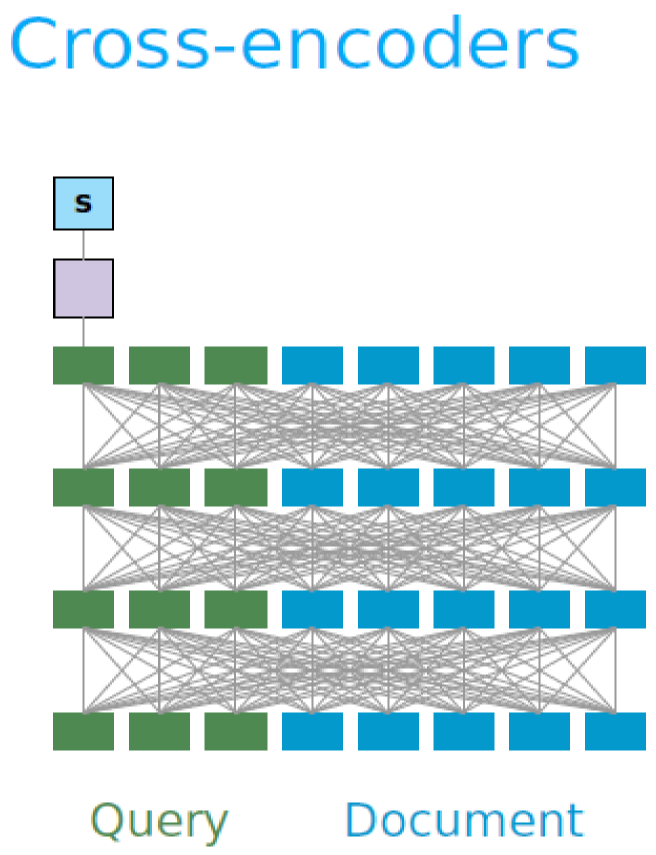
Cosine Similarity



ColBERT



Cross-encoder



Retrieving with reranker

Aspect	Cross-Encoder	ColBERT	Cosine Similarity (Bi-Encoder)
Architecture	Full interaction (query + doc jointly encoded)	Late interaction (token-level dot products)	Separate encodings (query & doc)
Speed	Slowest	Medium	Fastest
Accuracy	Highest	High	Lower
Scalability	Low	Moderate	High
Latency	High (per pair processing)	Medium (token matching cost)	Low (instant vector search)

Summarization with RAG

Gemini Prompt-based

BART-large Task-based

Text is summarized into concise bullet points using either Google Gemini or a Hugging Face model (BART-large-cnn)

Gemini is used as the primary backend, selected for its strong language modeling and ability to follow custom prompts

Zero shot

Directly asks the model to summarize without examples

Few shots

Provides examples of input-output pairs to guide the model

Chain of Thoughts

Guides the model to reason through key ideas before summarizing

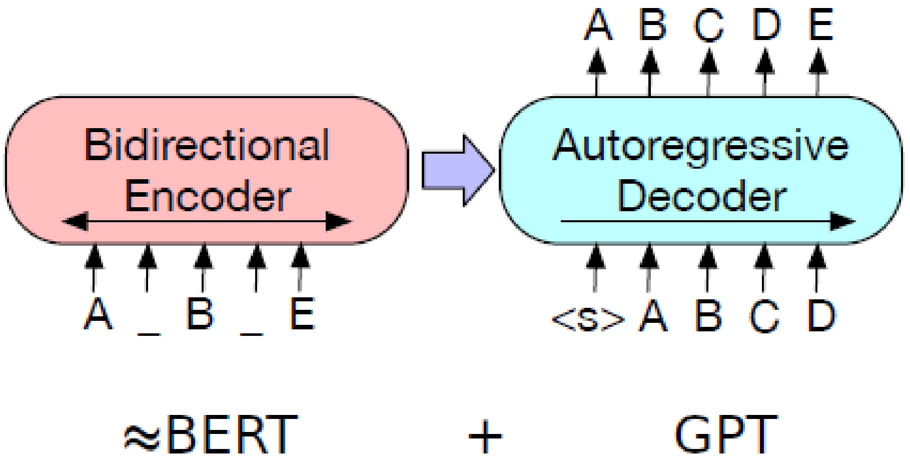
If Gemini fails or is unavailable, the system falls back to a local Hugging Face model, ensuring robust summarization even offline.

Why is BART a good candidate for this task?

BART can be useful in cases where data extraction is weak, as it is capable of filling in missing words and reconstructing context

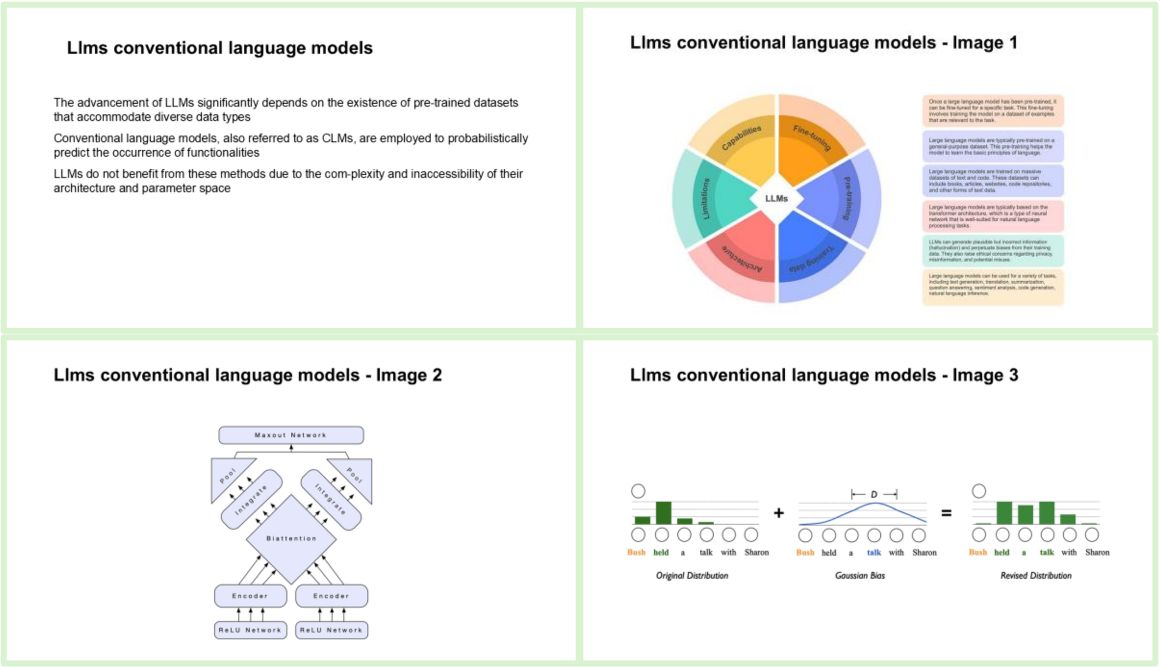
BART

- Text infilling
- Sentence shuffling
- Token masking
- Token deletion
- Document rotation

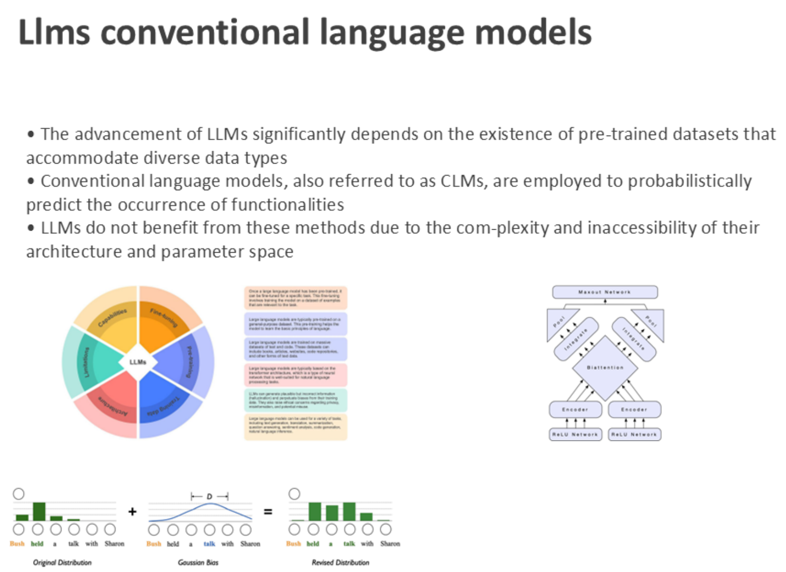


Evolutionary Layout Optimization (ELO)

ELO optimizes PowerPoint slide layouts from predefined content, maximizing presentation quality on *readability*, *hierarchy*, *image integration*, *contrast*, *balance*, and *constraints*.



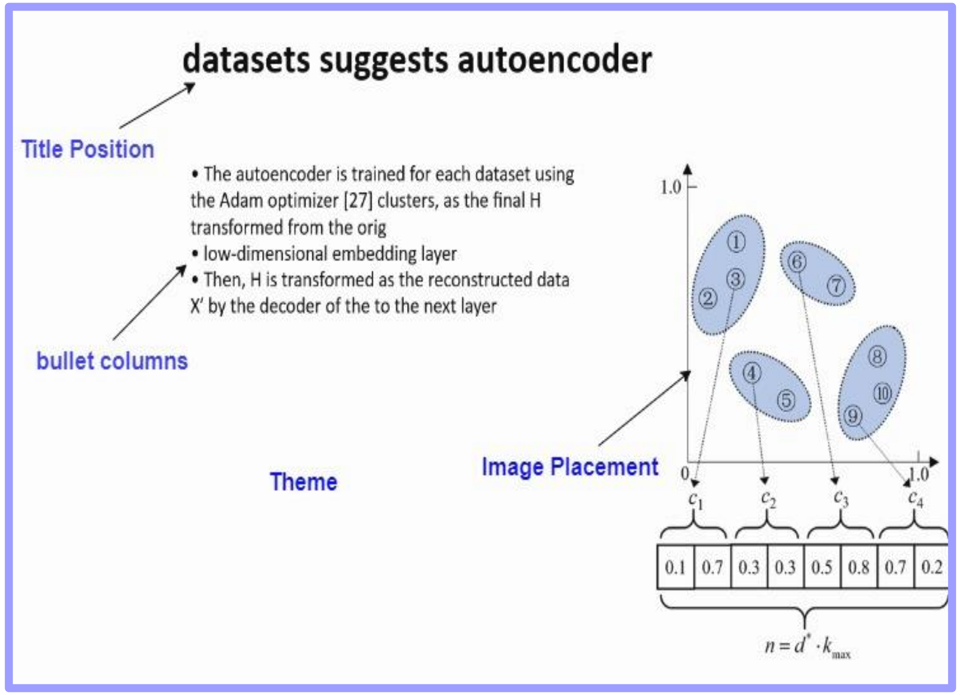
Before



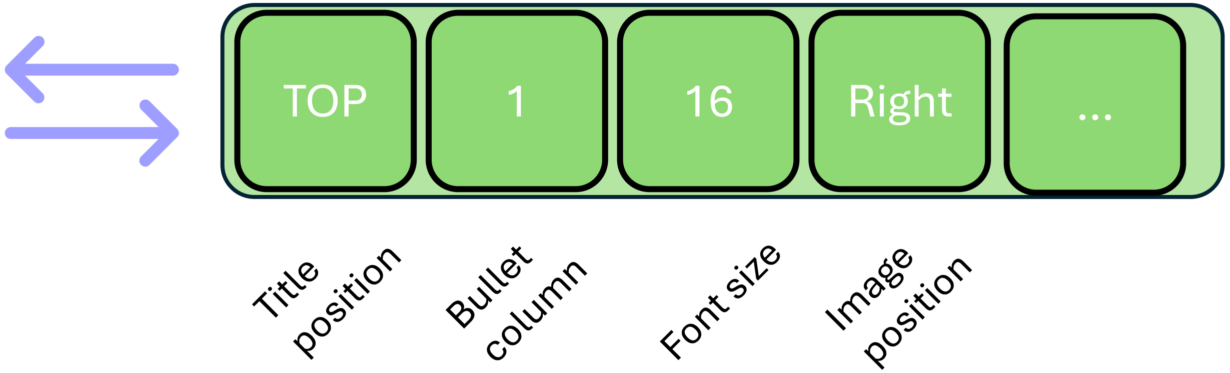
After

Presentation Phenotype/Genotype

Phenotype



Genotype



Each slide's layout is represented as a set of discrete variables called "genes" encoding design choices

Mathematical formulation

- Each slide is composed of
  - T title, B bullet points, I images
  - $S = \{T, B = \{b_1, b_2, , \ldots, b_{max}\}, I = \{i_1, i_2, , \ldots, i_m\})$
- Genotype (layout encoding)
  - $L = \{f_t, f_b, c_b, p_i, l_i, \; s_i, m, b_{it}, p_t, c_t, c_b, \theta\}$

Gene	Description
f_t	Title font size
f_b	Bullet font size
c_b	Number of bullet columns
p_i	Image position
l_i	Image layout type
s_i	Image size
m	Margin size
b_{it}	Image-text balance
p_t	Title position
c_t	Text color
c_b	Background color
θ	Theme

Mathematical formulation

- Content features (content profile K)
  - The content S is also characterized by measurable properties.
  - $K = \{|B|, \overline{w}_b, \dots, c_i\}$
- Objective function (fitness F)

$$F(L,K) = \sum_{i=1}^{10} \lambda_i \, \boldsymbol{f}_{\boldsymbol{i}}(\boldsymbol{L},\boldsymbol{K})$$

- $\lambda_i \in [0,1]$ is a tunable weight for each component
- The fitness function depend on both the layout L and the content it holds K, as K is what makes ELO content-aware.

Feature	Description
$\|B\|$	Number of bullet points
$\|I\|$	Number of images
$\overline{w_b}$	Avg. words per bullet
w_{total}	Total word count
w_T	Title word count
δ_i	Boolean for image presence
O_I	Dominant image orientation
δ_{mix}	Mixed orientations
δ_{large}	Presence of large images
r_{ti}	Text-to-image ratio (e.g., "balanced")
d_c	Content density
c_i	Image complexity

Mathematical formulation

- K allows the layout optimizer to **adapt** the design of each slide to match its content.
  - A slide with many bullet points needs more space or multiple columns
  - A slide with large images must allocate enough room without crowding text
  - A slide with no images shouldn't waste space with empty placeholders
- K ensures that layout quality is not judged in isolation but based on how well the layout fits the actual content on the slide.

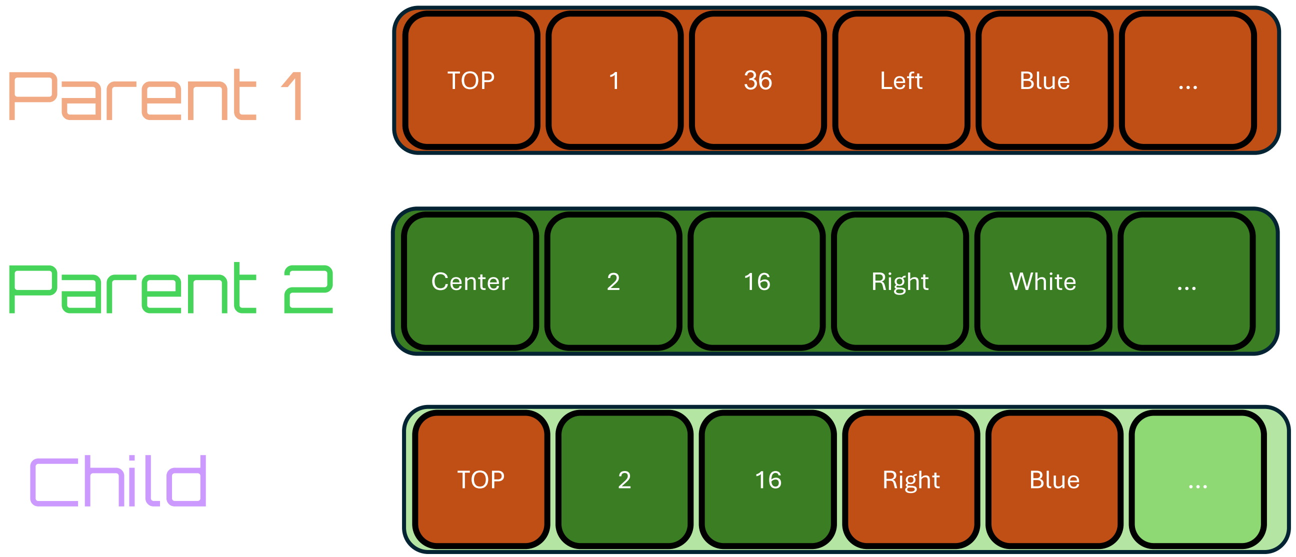
- Including K help the system avoids overfitting to fixed layout heuristic

- Evolutionary operators (e.g., mutation) can be smarter when they know the slide contains dense text or multiple visuals — allowing for **adaptive evolution**.

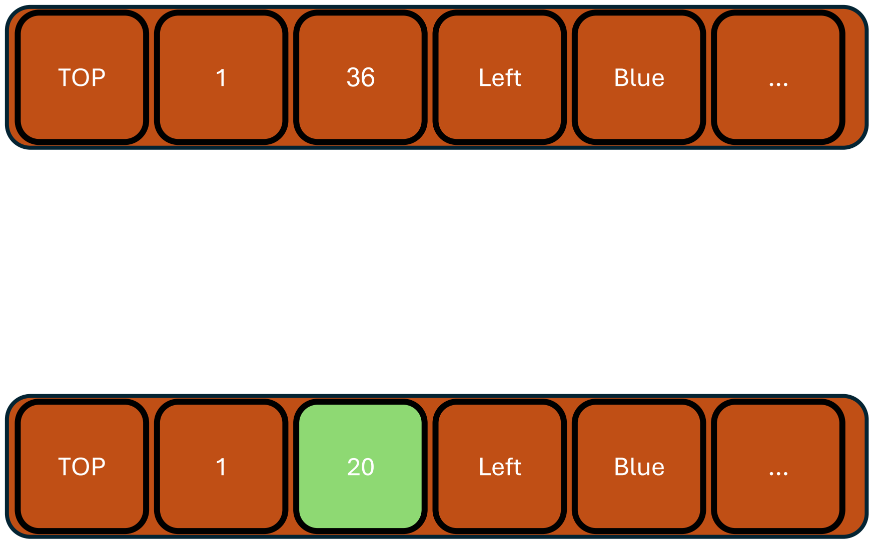
Mathematical formulation

Quality	Description	How to measure?
f_1	Text Readability	Penalty for small font sizes, overlapping elements, and low contrast text
f_2	Bullet Management	Evaluates column count, spacing, and bullet alignment based on number/length of bullets
f_3	Image Handling	Penalizes distorted/resized images or overlap with text; rewards proper scaling
f_4	Visual Balance	Compares horizontal and vertical distribution of content regions (centroid analysis)
f_5	Color Contrast	Computes luminance difference between text and background colors
f_6	Consistency	Checks font sizes, alignment, and color consistency with theme settings
f_7	Content Appropriateness	Matches layout choice (e.g., number of columns) to content properties (bullet count, density)
f_8	Accessibility	Penalizes low-contrast text, small fonts, or crowded content
f_9	Image-Text Balance	Computes ratio of occupied text/image areas vs. total slide area
f_{10}	Boundary Constraints	Applies hard penalties for content outside margins or overlapping bounding boxes

Evolutionary operators



Uniform crossover



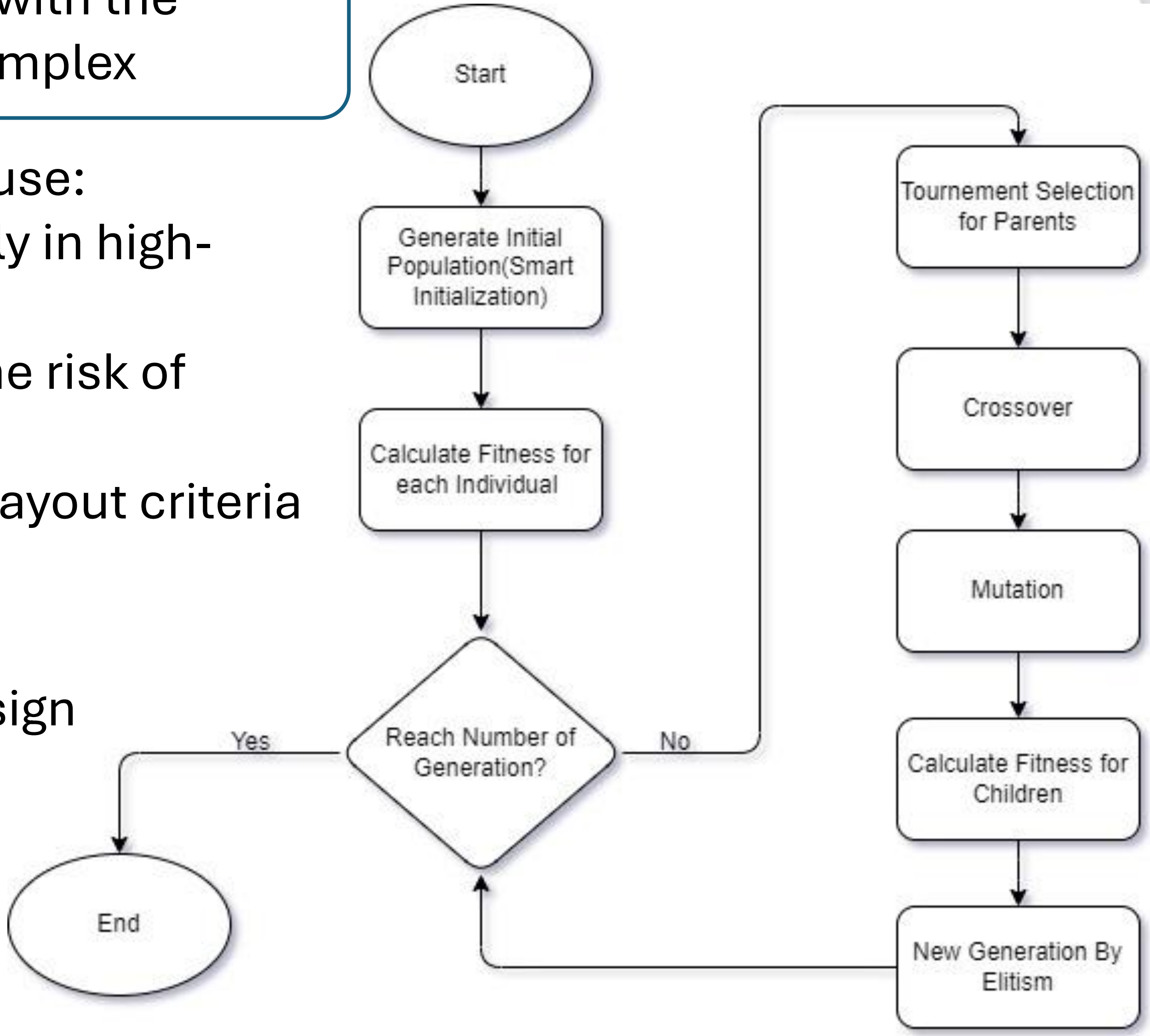
Flip mutation

GA for ELO

Simple rules or brute-force can't cope with the layout search space—too large, too complex

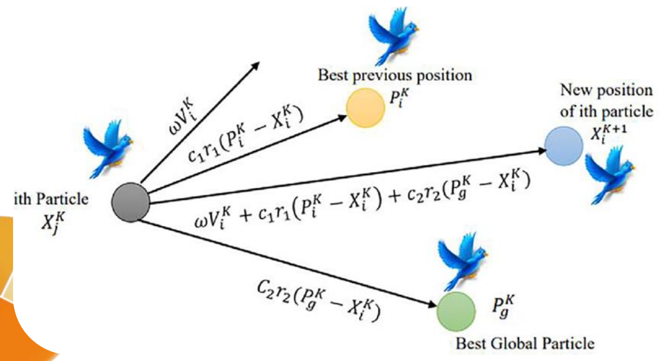
- Genetic Algorithm (GA) is ideal because:
  - It searches globally and efficiently in high-dimensional, discrete spaces.
  - It maintains diversity, reducing the risk of getting stuck in local optima.
  - It adapts well to multi-objective layout criteria (fitness function).

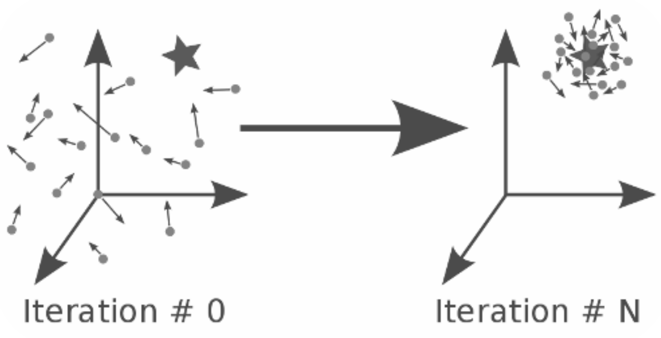
- Empirically proven for layout and design problems.

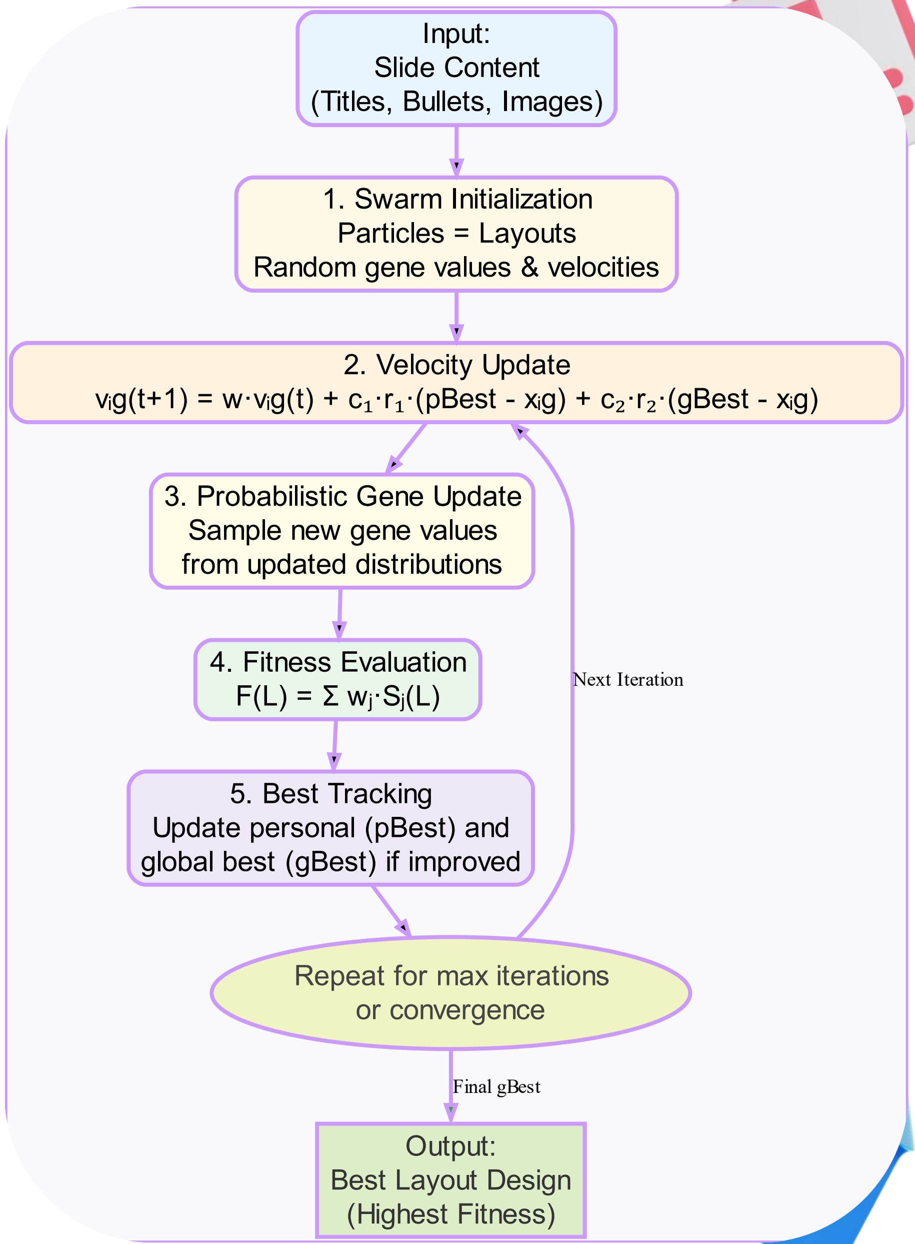


PSO for ELO

- PSO models a "swarm" of particles exploring layout space, inspired by birds flocking towards food.
  - Each particle represents a possible slide layout.
  - Particles share information:
    - They move in the direction of their own best found solution (personal best)
    - They are also pulled toward the global best solution found by the swarm.
  - Layout updating in discrete design:
    - Each gene (e.g., title position, image layout) is updated based on a probability distribution, influenced by local/global bests.

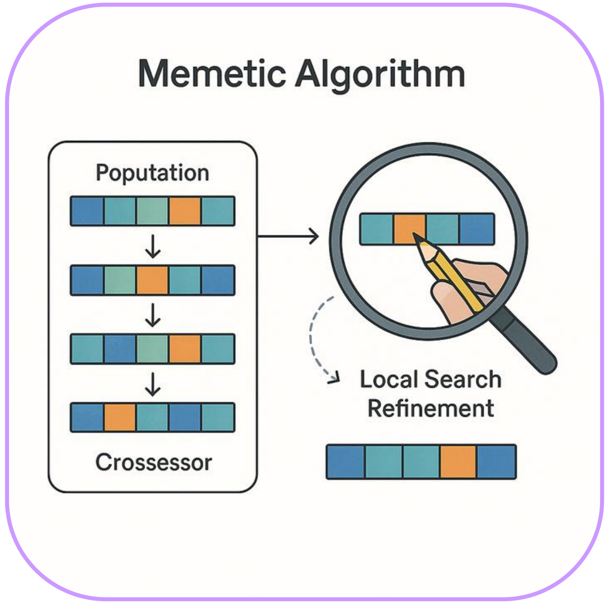


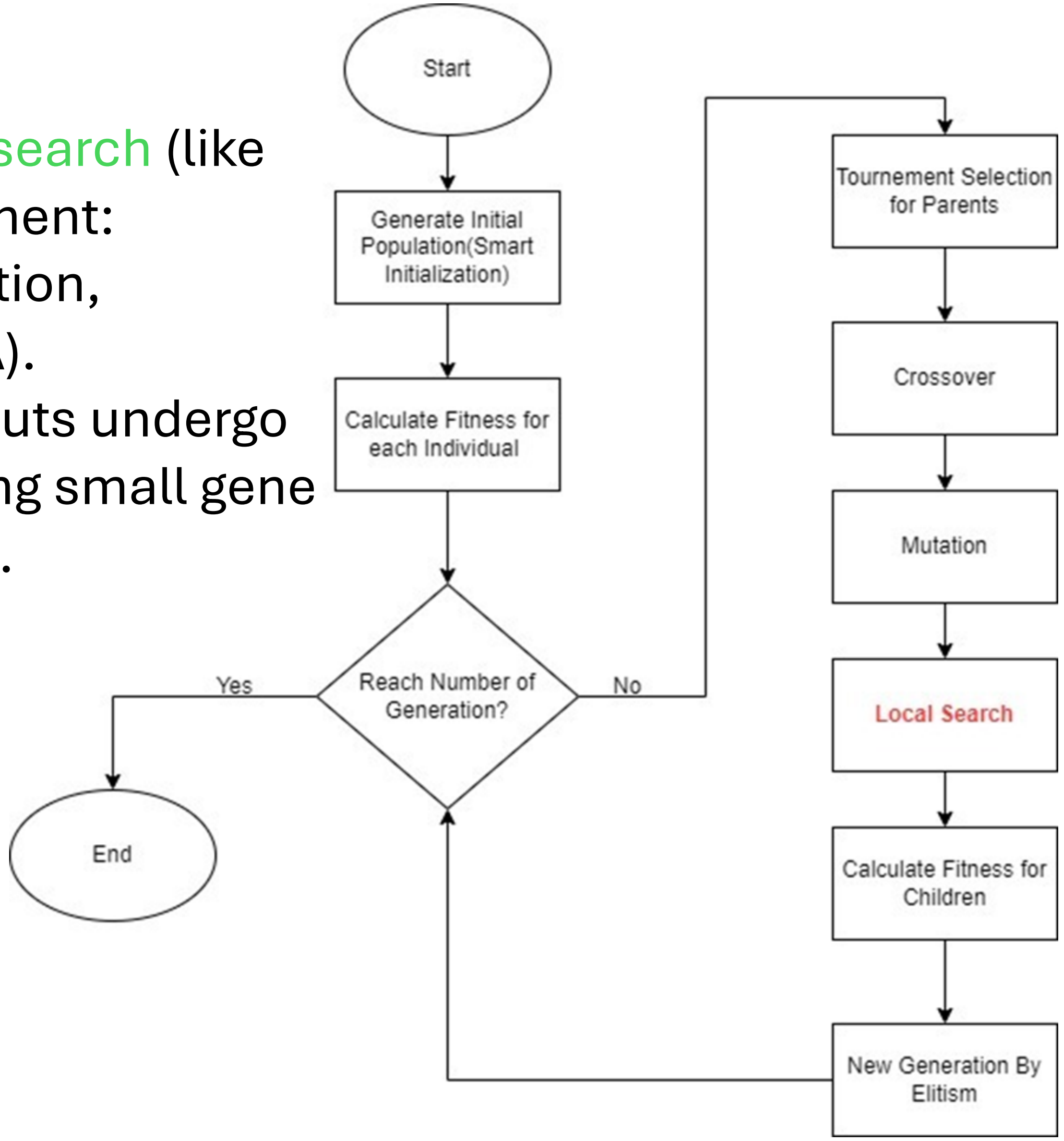




Memetic for ELO

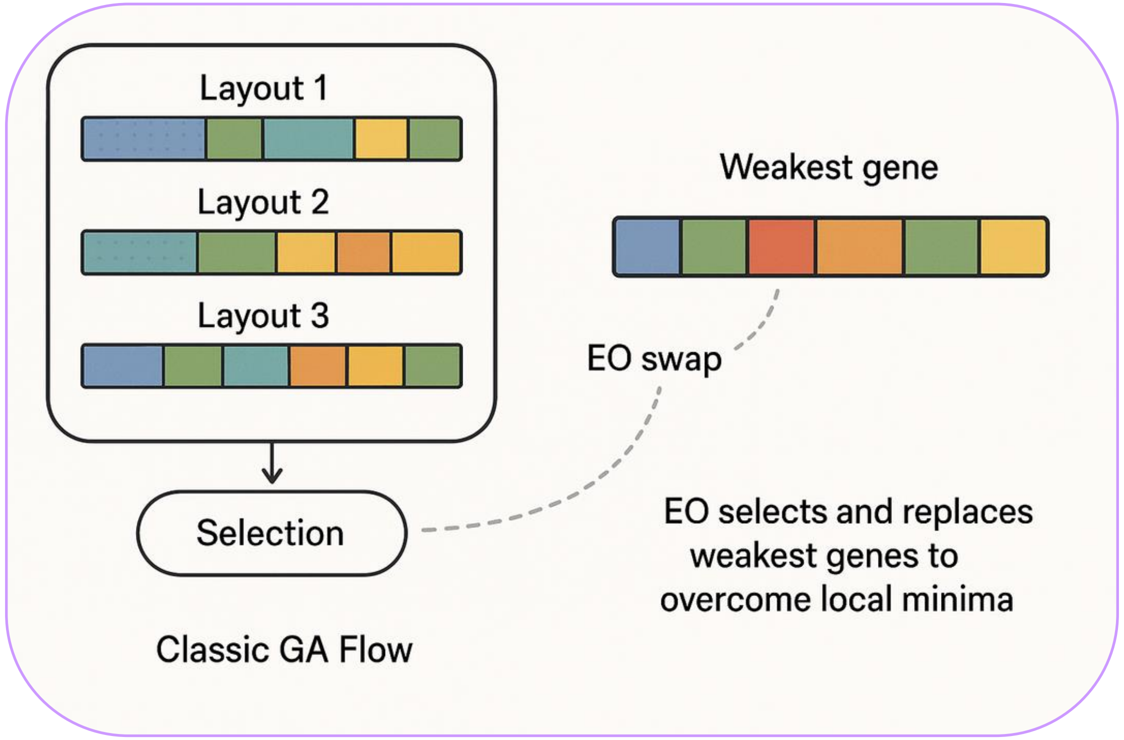
- Memetic Algorithm combines global search (like GA) with local search for extra refinement:
  - Population evolves through selection, crossover, and mutation (as in GA).
  - After each generation, select layouts undergo "local search" (hill climbing), trying small gene tweaks to further increase fitness.





Hybrid EO-GA for ELO

- Hybrid EO-GA alternates standard GA evolution with τ-Extremal Optimization (EO):
  - GA phase explores broadly, combining and mutating layouts as usual.
  - EO phase targets the worst-performing genes (“weakest links”) in top layouts, probabilistically replacing them to break out of local optima.
  - Uses a heavy-tailed probability distribution controlled by parameter τ.



Evaluation

Rank	Algorithm	Fitness Score	Time (s)	Convergence	Stability	Overall Score
1	Memetic	176.80±97.38	1.32±0.49	**0.844**	0.850	**0.882**
2	GA	**178.60±94.34**	**0.35±0.09**	0.500	0.750	0.821
3	PSO	172.33±101.33	0.70±0.3	0.794	**0.926**	0.810
4	Hybrid GA-EO	175.00±100.29	11.18±1.89	0.500	0.800	0.658

The table shows that GA achieves the highest average fitness and fastest runtime, consistently outperforming other methods in solution quality and efficiency, making it the most effective choice for slide layout optimization.

Performance Analysis by Slide Type

GA, Hybrid GA-EO, and Memetic all achieved a fitness score of 135.00 across all slide types.

GA achieved the fastest runtime at 0.270s

On complex, high-density slides, GA scored a modest 17.99 with significant result variation.

Evaluation

We use ROUGE-1 to ROUGE-4, ROUGE-L, and ROUGE-S to measure similarity to reference summaries

We use LLM-based metrics (e.g., G-Eval via Gemini and LLaMA-2) to score summaries on coherence , relevance, and fluency

We use BERTScore for semantic similarity and CLIP for text-image alignment to assess language and visual relevance

Metric Name	Purpose	Model / Method	Output Score
ROUGE-1	Unigram (word-level) overlap	N-gram F1-measure	[0, 1]
ROUGE-2	Bigram overlap	N-gram F1-measure	[0, 1]
ROUGE-3	Trigram overlap	N-gram F1-measure	[0, 1]
ROUGE-4	Four-gram overlap	N-gram F1-measure	[0, 1]
ROUGE-L	Longest common subsequence	LCS F1-measure	[0, 1]
ROUGE-S	Skip-bigram overlap	Skip-bigram F1-measure	[0, 1]
ROUGE-SL	ROUGE-L with length normalization	Adapted ROUGE-L formula	[0, 1]
G-Eval (LLM, Gemini)	LLM-based human-aligned judgmen	Gemini LLM (Google)	1–5 (per dimension)
G-Eval (LLM, Llama)	LLM-based human-aligned judgment	Llama-2 LLM (Meta)	1–5 (per dimension)
Text-Figure Relevance	Multimodal slide text/image alignment	CLIP similarity	[0, 1]
BERTScore	Contextual semantic similarity	Transformer (BERT/RoBERTa)	[0, 1]
BLEURT	Learned quality metric, human-	Fine-tuned BERT on ratings	[-1, 1]

Evaluation

Low ROUGE scores (e.g., ROUGE-1: 0.089, ROUGE-2: 0.013) indicate minimal word overlap with reference texts.

G-Eval (Gemini) excelled in fluency but showed uneven scores across other aspects.

while G-Eval (LLaMA) delivered more balanced results, indicating better overall alignment with human judgment.

BERTScore achieved the highest score (0.5180), reflecting strong semantic similarity with reference content.

Metric	Score (Average over all PDFs/slides)
ROUGE-1 (F1)	0.08966
ROUGE-2 (F1)	0.013567
ROUGE-3 (F1)	0.00107
ROUGE-4 (F1)	0.000267
ROUGE-L (F1)	0.05533
ROUGE-S (F1)	0.0034
ROUGE-SL (F1)	0.0125
G-Eval (Gemini)	- Coherence: 3.55 - Consistency: 2.50 - Fluency: 4.47 - Relevance: 2.86
G-Eval (Llama)	- Coherence: 3.71 - Consistency: 3.31 - Fluency: 3.82 - Relevance: 3.70
Text-Figure Relevance	0.2278
BERTScore (F1)	0.5180
BLEURT	0.2787

TEAM

Ahmed M. Makboul

Ahmed Mostafa

Hossam Ahmed

Omar Ayman

Ahmed Bassiouny

Ahmed Ismail

Thanks